



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**Návrh simulačního modelu pohybu autonomního mobilního robotu
v konvoji**

Autonomous mobile robot simulation model design for convoy vehicles

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Alexandr Capovsky

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Stanislav Věchet, Ph.D.

BRNO 2017

Zadání bakalářské práce

Ústav: Ústav automatizace a informatiky
Student: **Alexandr Capovsky**
Studijní program: Strojírenství
Studijní obor: Aplikovaná informatika a řízení
Vedoucí práce: **doc. Ing. Stanislav Věchet, Ph.D.**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh simulačního modelu pohybu autonomního mobilního robotu v konvoji

Stručná charakteristika problematiky úkolu:

Návrh simulačního modelu pohybu autonomního mobilního robotu v konvoji vozidel. Cílem je realizace simulačního modelu pro prostředí ROS umožňující realizaci konvoje silničních vozidel zahrnující objíždění překážek či ztrátu kontaktu se sledovaným vozidlem.

Cíle bakalářské práce:

- 1) Seznamte se s aktuálním stavem frameworku ROS.
- 2) Prostudujte tvorbu simulačních modelů a jejich implementaci v rámci frameworku ROS.
- 3) Navrhnete vhodný simulační model pohybu autonomních vozidel v konvoji.
- 4) Navržený systém otestujte.

Seznam literatury:



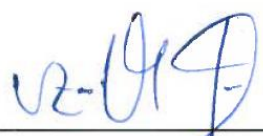
THRUN Sebastian, BURGARD Wolfram, and FOX Dieter. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series). Intelligent robotics and autonomous agents. The MIT Press, August 2005.

CHOSSET Howie, LYNCH Kevin M., HUTCHINSON Seth, KANTOR George, BURGARD Wolfram, KAVRAKI Lydia E., and THRUN Sebastian. Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents). The MIT Press, June 2005.

ROS.org. ROS.org | Powering the world's robots. [online]. 2.11.2016 [cit. 2016-11-02]. Dostupné z: <http://www.ros.org/>.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17.

V Brně, dne 7. 11. 2016

		
_____ doc. Ing. Radomil Matoušek, Ph.D. ředitel ústavu		_____ doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

ABSTRAKT

Práce se zabývá tvorbou simulačního modelu pro řízení autonomních vozidel v konvoji.

ABSTRACT

This thesis deals with creating of model simulator for autonomous convoy vehicles.

KLÍČOVÁ SLOVA

Autonomní řízení, konvoj, robotika, optimální trajektorie, odometrie.

KEYWORDS

Autonomous driving, convoy, robotics, optimal trajectory, odometry.

BIBLIOGRAFICKÁ CITACE

CAPOVSKY, Alexandr. *Návrh simulačního modelu pohybu autonomního mobilního robotu v konvoji*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 44 s. Vedoucí bakalářské práce doc. Ing. Stanislav Věchet, Ph.D.

PODĚKOVÁNÍ

Děkuji vedoucímu mé bakalářské práce, doc. Ing. Stanislavu Věchetovi, Ph.D., za odborné vedení, podnětné připomínky a čas, který mi věnoval v průběhu vypracovávání této práce.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením doc. Ing. Stanislava Věchet, Ph.D. s použitím literatury uvedené v seznamu literatury.

V Brně dne 1. 3. 2017

.....

Alexandr Capovsky

OBSAH

1	ÚVOD.....	15
2	AUTONOMNÍ ŘÍZENÍ	17
2.1	Strojové vidění.....	18
2.1.1	Lidar (Light Detection And Ranging)	19
2.1.2	Ultrazvukové senzory	19
2.1.3	Kamera.....	20
2.2	Komunikace.....	21
3	LOKALIZACE	22
3.1	Mobilní robotika bez lokalizace	24
3.2	Globální a lokální lokalizace	24
3.3	Odometrie	24
3.3.1	Model „tříkolka“	26
3.3.2	Ackermanova geometrie řízení.....	27
3.4	Porovnání lokalizačních metod	29
3.5	ROS (Robot Operating System)	30
4	SIMULAČNÍ MODEL	31
4.1	Princip fungování modelu	31
4.2	Popis auta.....	31
4.3	Sestavení řídicích bodů.....	33
4.4	Řízení slave vozidla.....	34
4.4.1	Fáze 1.....	35
4.4.2	Fáze 2.....	36
4.4.3	Fáze 3.....	36
4.4.4	Fáze 4.....	36
4.4.5	Fáze 5.....	37
4.5	Testování	37
5	ZÁVĚR	41
6	SEZNAM POUŽITÉ LITERATURY.....	43
7	SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK	44

1 ÚVOD

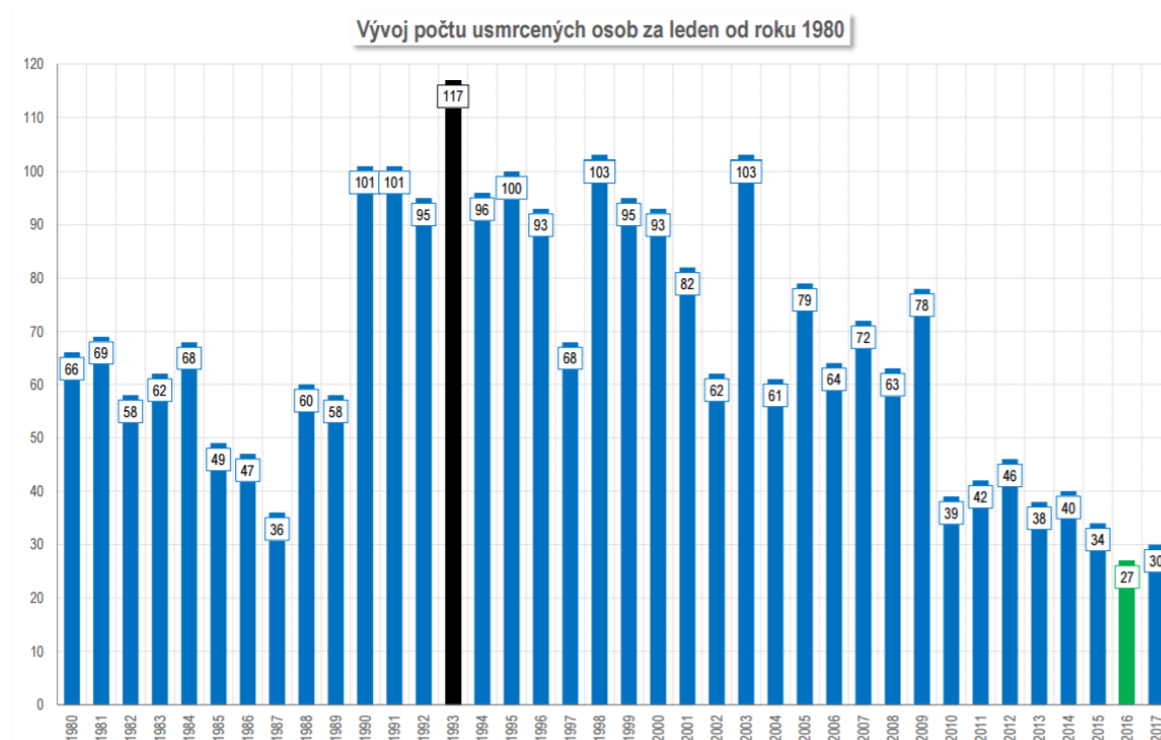
Automatizace je obor, který v dnešní době ovlivňuje mnoho technických disciplín. V minulosti byla automatizace využívána hlavně ve strojírenství, kde nahrazovala práci operátorů ve výrobě. Nyní má vliv i na každodenní činnosti člověka. Automatické zhasínání světel či robotický vysavač jsou jedny z mnoha příkladů, kde je automatizace využívána i v domácnosti. Nová oblast, kde je použita automatizace, je autonomní řízení vozidel.

Samostatné řízení vozidel zde existuje několik let. V dolech, na poli, ve skladech, se vozidla pohybují samostatně, ovšem vozidla v silničním provozu jsou úlohou, která je, co se týče složitosti náročnější.

Bakalářská práce se zabývá návrhem autonomního konvoje. Popisuje jednotlivé procesy, které umožňují udržet konvoj v chodu, a následně se bude zabývat jeho řízením. Cílem této práce je sestavení simulačního modelu, díky němuž lze navrhnout model řízení vozidel. Hlavní problém sledování více vozidel jedoucích v řadě je ten, aby nedocházelo ke zkracování dráhy při změně pohybu vedoucího vozidla. Úkolem práce je tento hlavní problém vyřešit.

2 AUTONOMNÍ ŘÍZENÍ

Autonomní řízení vozidel je technologií, která do značné míry změní způsob silniční dopravy. Slibuje větší bezpečnost, rychlost a efektivitu. Z bezpečnostního hlediska se situace na silnicích zlepšuje, jak ukazuje následující graf.

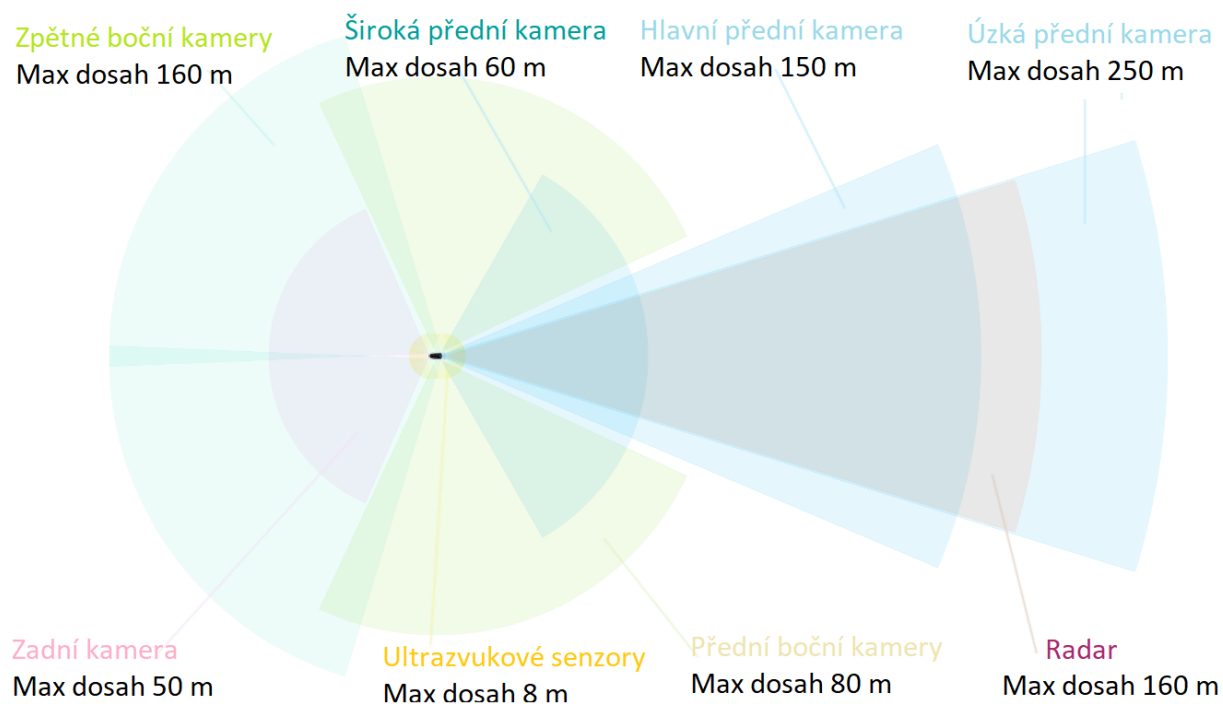


Obr. 1: Vývoj počtu usmrcených osob za leden od roku 1980 [1]

Přesto je podle statistik [1] více než 90 % nehod zapříčiněno lidskou chybou. Mnoho společností, jako Tesla, Volvo, Google nebo Mercedes, nyní investuje miliardy dolarů do vývoje autonomních systémů, které by časem měly nahradit lidského řidiče. Kromě politických a ekonomických problémů jsou zde i problémy technické.

V dnešní době jsme ve stádiu poloautomatizace. Volvo představilo svůj projekt SARTRE. Jedná se o konvoj vozidel, kdy je vedoucí vozidlo řízené člověkem, za který se pak mohou ostatní virtuálně připojit. Tento projekt se velmi podobá projektu, kterým se zabývá tato bakalářská práce.

Automobilka Tesla se snaží dosáhnout plné automatizace v dopravě. Nejnovější verze autopilota využívá 8 kamer, 1 radarový a 12 ultrazvukových senzorů. Detailně jsou popsány v následujícím obrázku.



Obr. 2: Popis senzorů v autopilotu Tesly [2]

Momentálně je tento systém určen pouze pro řízení na dálnicích při maximální rychlosti 110 km/h. Po okresních komunikacích je autopilot schopen jet rychlostí 36 km/h a to vše za kontroly řidiče, který je zodpovědný za případné škody.

2.1 Strojové vidění

V autonomním řízení máme 2 důležité aspekty, které jsou rozhodující pro správné fungování systému:

- 1) strojové vidění
- 2) lokalizace

Strojové vidění je důležité pro sběr dat. Při autonomním řízení vozidel je nutné vidět nejen cíl cesty, ale i případné překážky. V této kapitole se práce bude věnovat popisu metodik a senzorů pro sběr dat.

2.1.1 Lidar (Light Detection And Ranging)

Lidar funguje na jednoduchém principu odrazu světla. Při osvětlení objektu, tak to, co člověk vidí, je světlo, které se k němu vrátilo. Rychlost světla lze určit velmi přesně – 299 792 458 m/s. Jednoduchou úvahou tedy dojdeme k rovnici:

$$s = \frac{c * t}{2} \quad (1)$$

kde s je vzdálenost mezi objekty, c je rychlost světla a t je čas.

Lidar, známý již z 60. let, byl dříve používán pro detekci ponorek a k dalším vojenským účelům. Také byl využíván pro tvorbu map ke geologickým a archeologickým účelům.

Nevýhoda Lidaru a důvod, proč nebyl použit v našem projektu, je cena a překvapivě i rozsah. Pro příklad vyberu model LMS100 – 10000 od firmy SICK.

Parametry:

- rozsah: 0,5–20 m
- pozorovací úhel: 270°
- skenovací frekvence: 25 Hz / 50 Hz
- cena: 96 000 Kč

Tento model by byl ideální v kombinaci s námi použitou kamerou. Jelikož potřebujeme měřit i blízké vzdálenosti, použili jsme jiné řešení. Nutno dodat, že Lidar nepoužívá ani firma Tesla ve svém řešení autopilota z důvodu ceny.[3]



Obr. 3: Lidar SICK LMS100 – 100000

2.1.2 Ultrazvukové senzory

Ultrazvukový senzor je zařízení, které měří vzdálenost objektu pomocí zvukových vln. Funguje na podobném principu jako Lidar, pouze místo světla používá zvukové vlny. I pro ultrazvukový senzor platí rovnice (1).

Ultrazvukové senzory mohou být ovlivněné teplotou a vlhkostí vzduchu. Také některé objekty odrážejí zvuk takovým způsobem, že je senzor nedokáže zachytit. Existují objekty, které dokážou zvuk absorbovat. Jednou z dalších nevýhod je delší doba odezvy a širší detekční paprsek. Je tedy nutné vzít v potaz tyto skutečnosti před použitím ultrazvukových senzorů v autonomním řízení.

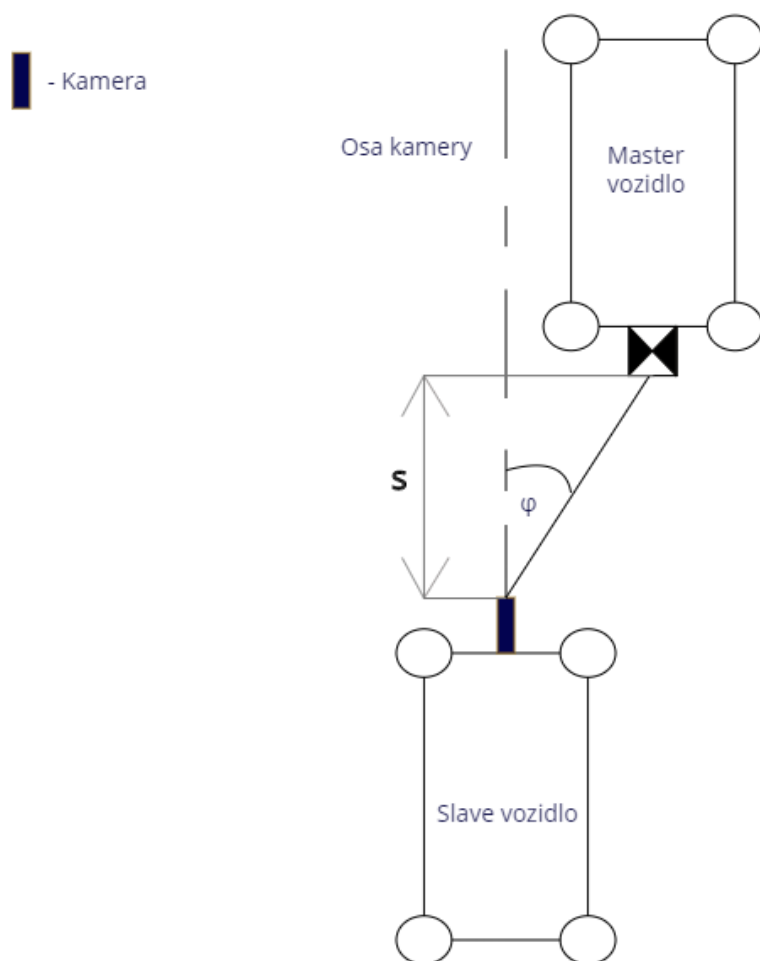
Hlavní výhodou je konstantní rozlišení v celém rozsahu měření kolem 3 mm a příznivá cena. Ultrazvukový senzor je vhodný pro kontrolu kamery a zjišťování nejbližšího okolí, kde by senzor sloužil pro objevení případné překážky, která by se mohla objevit v naplánované trase.[4]

2.1.3 Kamera

V projektu byla použita pouze jedna kamera pro detekci auta. Auto bylo detekováno pomocí značky Obr. 11. Díky práci kolegy Patrika Kaury[14] byly poskytnuty 2 parametry, které jsou zobrazeny v Obr. 4.

- úhel natočení φ značky vůči ose kamery
- vzdálenost s

Byla použita kamera Microsoft LifeCam HD-3000. Dosah kamery, který dokáže detekovat značku je 200–600 mm při frekvenci 25 snímků za sekundu. Dosah kamery byl zjištěn experimentálně. Je klíčové znát tyto údaje pro realistické naprogramování simulačního modelu.



Obr. 4: Schéma konvoje

2.2 Komunikace

Detailní popis komunikace včetně řídicích skriptů je popsána v práci kolegy Hynka Buchty [15]. V této kapitole je uveden stručný popis komunikace. Jakým způsobem je implementována a důležité údaje pro správnou tvorbu simulačního modelu.

V našem případě komunikace mezi kamerou a Raspberry PI modelem 3 protokolem je prováděna protokolem USB. Nároky na komunikaci nejsou veliké. Robotický konvoj obsluhuje pouze jednu kameru, která posílá 25krát za vteřinu přibližně 1 kB. Raspberry PI model 3 má zabudované USB 2.0 porty, které zvládají dostatečnou přenosovou rychlost, a sice 30 MB/s. Pro sofistikovanější řešení lze použít sběrnici CAN, popřípadě jiné sběrnice na bázi Ethernetu např.: VISA.

Vyhodnocování

Pro autonomní řízení je klíčová nejkratší reakční doba. V reálném projektu je nutné, aby vyhodnocení situace a předání informace probíhalo „real-time“. Dlouhá odezva může zapříčinit havárii vozu. Procesor v Raspberry Pi model 3 je 1.2GHz 64bitový čtyřjádrový ARM Cortex-A53, který výkonnostně stačí přijímat data, vyhodnocovat cestu a posílat data do motoru.

3 LOKALIZACE

V této kapitole bude popsána lokalizace robotů a rozdělení robotů z hlediska automatizace provozu. Budou vysvětleny jednotlivé postupy a výhody různých metod lokalizace.

Robot je stroj, který částečně pracuje sám. Robotika je interdisciplinární obor, který se zabývá konstrukcí, řízením a využitím robotů. V dnešní době se robotika vyvíjí velmi rychle a vznikají nové typy robotů. Tato bakalářská práce se zabývá výhradně mobilními roboty.

Mobilní robot dokáže měnit svou pozici. Do této kategorie lze zařadit auto, dron či neobvyklého robota, který svým pohybem připomíná krávu a využívá ho armáda pro přenášení materiálu po bitevním poli. Zvláštní skupina robotů – humanoidi, kteří nahrazují práci číšníku a průvodců a tvarem a způsobem pohybu připomínají člověka.



Obr. 5: Robot LS3 od firmy DARPA [11]

Roboty můžeme rozdělit z hlediska autonomie:

- Dálkově ovládané (*teleoperované*), kdy jediným úkolem robota je co nejlépe poslechnout pokyny živého operátora.
- *Autonomní* roboti pracující samostatně.
- *Semi-autonomní* roboti, částečně pracují sami. Cílem je ulehčit jednoduché rutinní činnosti člověku, který by měl společně s robotem spolupracovat, což je i náš případ. Konvoj pracuje v semi-autonomním režimu [5]

Velmi důležitá informace je, v jakém prostředí se robot pohybuje. Ve statickém prostředí, které je stálé a neměnné v čase je pohybující robot v prostoru sám.

Naproti tomu dynamické prostředí se mění v čase. Pro názornou ukázkou – autonomní vysavače musí fungovat v dynamickém prostředí, zatímco při návrhu průmyslového robota lze předpokládat prostředí statické.

V bakalářském projektu se počítá s prostředím statickým. V praxi to znamená, že se nepředpokládá interakci s jinými objekty či událostmi, které by zapříčinily nemožnost slave vozidla projet předem vymezenou dráhu. Robot není schopen rozeznat jakékoliv překážky. Při nárazu master vozidla do zdi pravděpodobně nedojde k nárazu slave vozidla díky skutečnosti, že slave vozidlo udržuje od mastera určitou bezpečnou vzdálenost, ale zaručit to nelze. Pokud by se signál neztratil a master vozidlo by pokračovalo v trase i po nehodě, slave vozidlo by následovalo master vozidlo a také by do zdi nabouralo. Zde vidím prostor pro zlepšení, jelikož prostředí, ve kterém se pohybují reálné auta, je jednoznačně dynamické.

3.1 Mobilní robotika bez lokalizace

Moderní způsob ovládaní robotů. Není zde nutné přesně zadávat, jakým způsobem se robot bude pohybovat. Stačí pouze určit cíl a robot se pomocí mechanického učení dostane na určenou pozici. Tento způsob pohybu ovšem postrádá nahlédnutí do nitra řídicího systému, což nemusí být vždy žádoucí. [5]

Příkladem takového projektu v mobilní robotice naleznete na této adrese: <http://janhuenermann.com/projects/learning-to-drive>, [12] kde autor naprogramoval řízení vozidel v neznámém terénu. Za běhu programu lze táhnutím myši přidat další překážky. Kód je zdarma sdílen na GitHubu k nahlédnutí.

3.2 Globální a lokální lokalizace

Pomocí globální lokalizace lze odhadnout absolutní polohu v prostoru bez ohledu na předcházející události. Příkladem takového systému globální lokalizace je systém GPS. Naproti tomu lokální lokalizace odhaduje změnu polohy robota na základě jeho předchozí polohy. Skládáním jednotlivých poloh lze získat průběh pohybu robota v čase.

V robotických aplikacích se pomocí pravděpodobnostního přístupu často kombinuje globální a lokální lokalizace. Příkladem použití je „kidnapped robot problém“. Kdy robotu je změněna jeho poloha, aniž by to jeho senzory byly schopné zjistit. Robotická aplikace, která využívá tuto robustní strategii lokalizace, by měla být schopná zjistit, že robot se nenachází v místě určené lokální lokalizací. Tedy pomocí globalizačních algoritmů zjistí novou aktuální polohu robota. Jakmile je poloha robota nalezena, její následné sledování je prováděno lokální lokalizací. [5]

Tato bakalářská práce využívá lokální lokalizaci, proto budou v dalších kapitolách popsány možné způsoby jejího provedení.

3.3 Odometrie

Odometrie je metoda relativní lokalizace založená na odhadu změny pozice a orientace kolového robota prostřednictvím údajů o otáčení jeho hnacích nebo běžných kol naměřených pomocí rotačního encodéru. Název metody vznikl složením řeckých slov *odos* (cesta) a *metron* (měřiti). [5]

V této práci zjišťují relativní lokalizaci pomocí kamery, která detekuje značku. Tato skutečnost nám ovšem nebrání v použití kinematických či dynamických modelů, pomocí kterých můžeme sestavit simulační program.

Pro určení odometrie, je nutné znát, jakým způsobem se robot dokáže pohybovat. Mobilní roboty lze rozdělit do těchto tří kategorií:

auto

Kategorie *auto* zahrnuje roboty, kteří dokážou měnit svou polohu natočením kola či nápravy a následnou jízdou vpřed či vzad. [6] Osobní automobily mají obvykle

zatačející kola přední nápravy, kvůli jednoduchosti mechanických převodů. Ovšem u strojů, které kladou vysoké nároky na manévrovatelnost, jsou zatačející kola u zadní nápravy. Příkladem takového stroje je vysokozdvizný vozík.

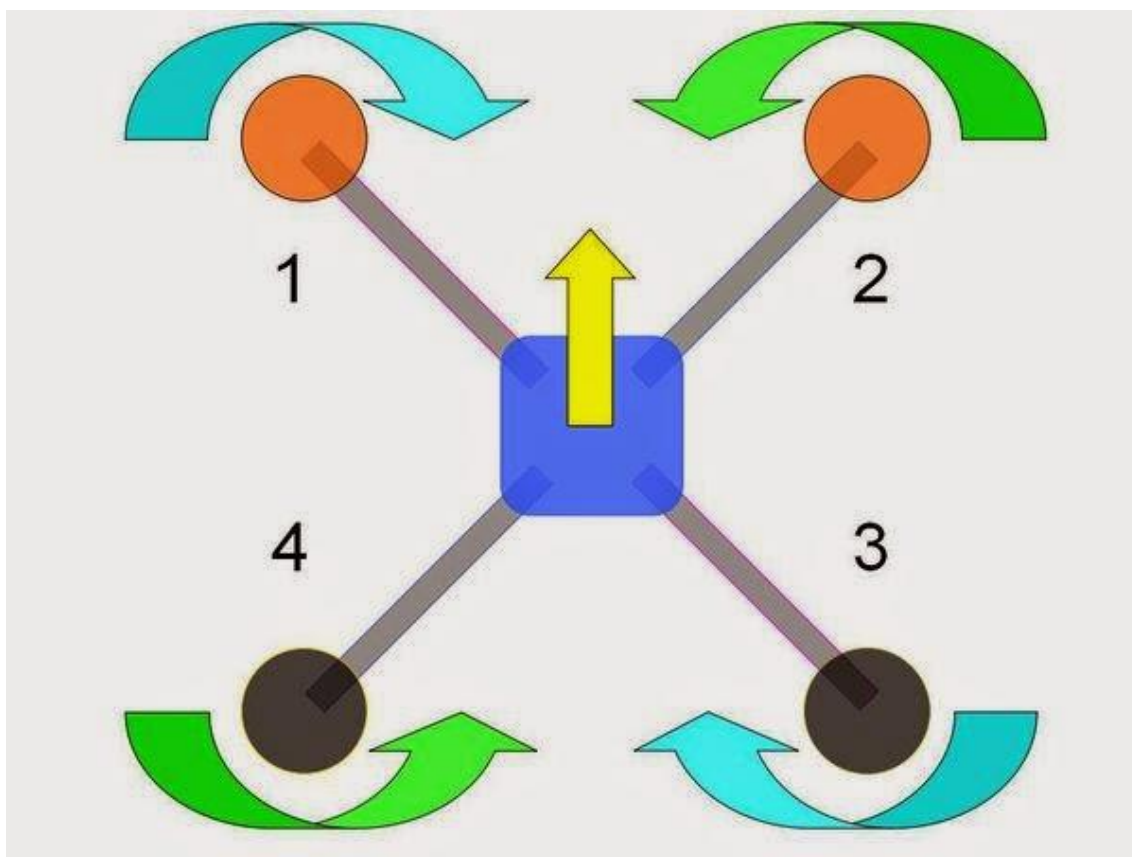
tank

Do kategorie *tank* patří všichni roboti, kteří jsou schopni se otáčet na místě, ale pohybují se pouze vpřed či vzad. Protože pásy se nemohou natočit, otáčení probíhá způsobem, kdy pásy se pohybují různou rychlostí. Součástí, která zajišťuje různý pohyb pásu, se nazývá diferenciál. [16]

všesměroví

Všesměroví roboti, se dokážou pohybovat všemi směry, mají 3 stupně volnosti. [6] Příkladem takového všesměrového robota je dron.

Dron, nejčastěji kvadrokoptéra, je létající stroj se 4 vrtulemi. Důvod, proč vznikl až teď, je prostý: toto uspořádání není schopno samotného letu bez kontroly počítače. Je zapotřebí řídicí jednotka vybavená gyroskopy a akcelerometry pro úpravu chodu motoru. Při řízení kvadrokoptéry se stejným směrem točí vždy protilehlé vrtule. Pro stoupání se motory zrychlí, pro klesání naopak zpomalí. Při pohybu určitým směrem např. dopředu. Zpomalí vrtule 1 a 2 zatímco zrychlí vrtule 3 a 4. Pomocí náklonu se dron začne pohybovat. Obdobný princip funguje při pohybu drona dozadu, doleva či doprava.



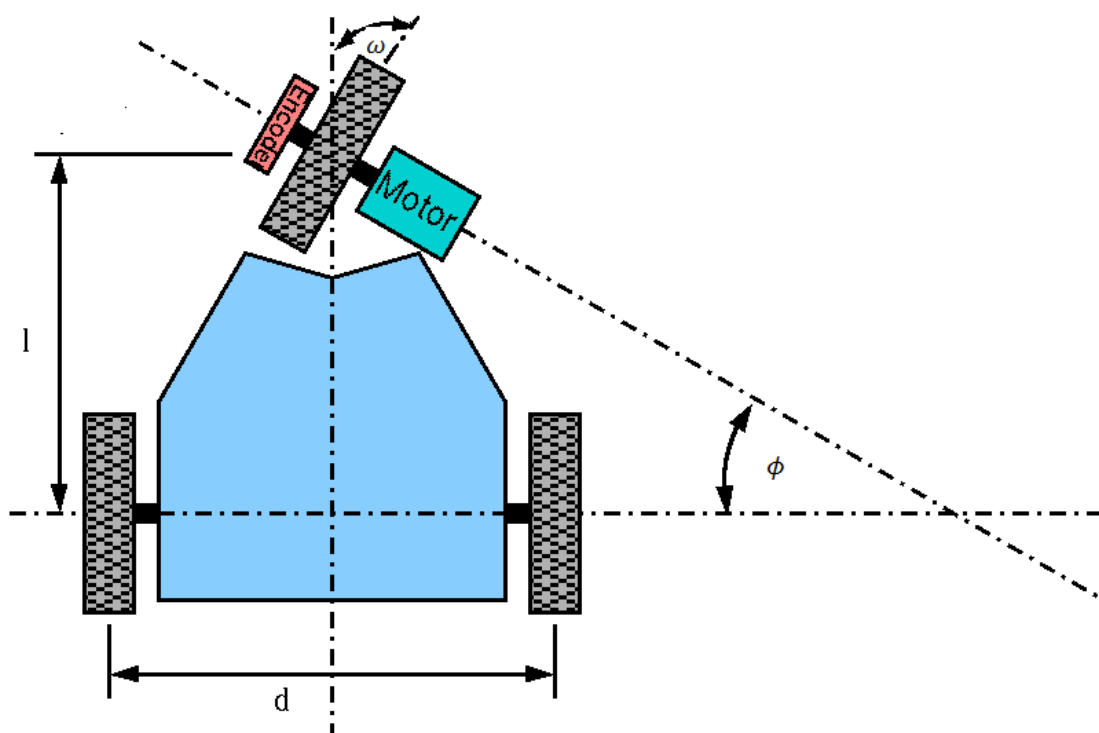
Obr. 6: Princip pohybu dronu [9]

Kde žlutá šipka naznačuje směr pohybu drona. Čísla označují pozici jednotlivých vrtulí a šipky u nich ukazují směr otáčení vrtule.

Obecně platí, že čím větší počet vrtulí, tím bezpečnější provoz na úkor obratnosti. Proto mají závodní drony zpravidla 3 vrtule, zatímco drony nesoucí drahé kamery mají vrtulí více. [9]

3.3.1 Model „tříkolka“

Důvod, proč je vysvětleno řízení auta na modelu tříkolky, bude zřejmý z kapitoly 2.3.2. Tříkolka je auto, které má 2 kola na vlečené nápravě a jedno řiditelné kolečko na přední nápravě. Model je zobrazen na následujícím obrázku.



Obr. 7: Model tříkolky [6]

Změnu pozice robota lze zjistit pomocí následujících rovnic.

Nejprve je důležité zjistit poloměr otáčení. Ten lze zjistit pomocí rovnice:

$$R = \frac{l}{\tan(\omega)} \quad (2)$$

kde R je poloměr otáčení, l je rozvor nápravy, tedy vzdálenost mezi přední a zadní nápravou a ω je úhel natočení kol.

Pomocí poloměru otáčení je zjištěna kružnice, po které se robot posouvá. Nyní záleží, jaké další informace o pohybu robota jsou k dispozici. Obecně platí pro úhlovou dráhu vztah:

$$\phi = \frac{dF * \cos(\omega)}{R} \quad (3)$$

Kde ϕ je úhlová dráha. dF je derivace síly působící na auto. V kapitole 4, je napsané jakým způsobem byla vypočtena úhlová dráha.

Dále je zapotřebí převést tento pohyb do kartézského souřadného systému. Změna pozice dx a dy se rovná:

$$dx = R - R * \cos(\phi) \quad (4)$$

$$dy = R * \sin(\phi) \quad (5)$$

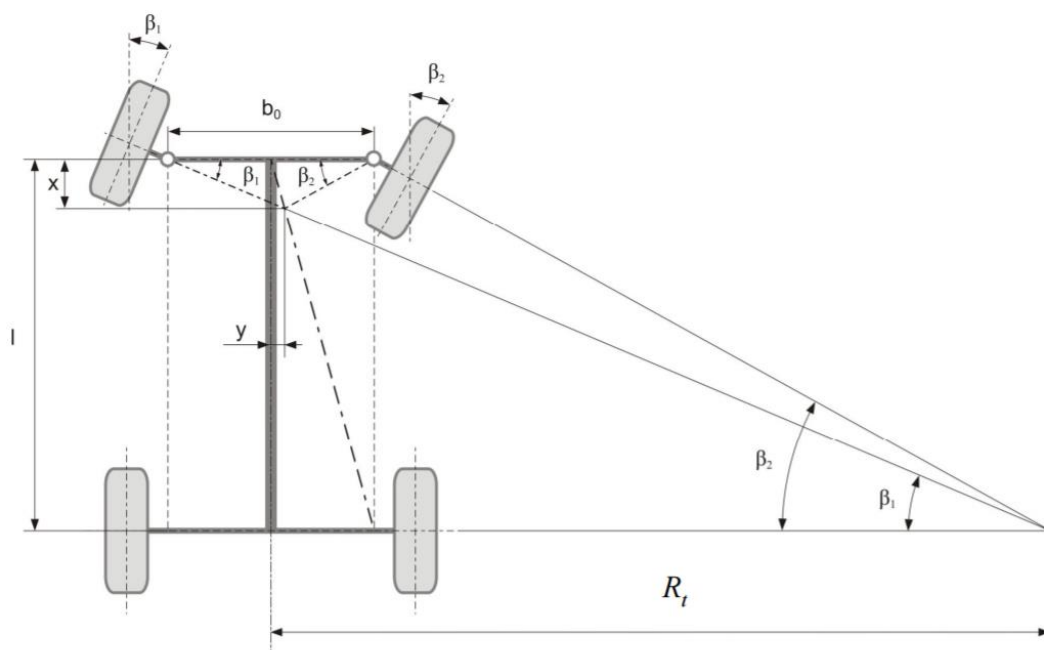
Manévrovací schopnosti auta jsou závislé na dvou parametrech. Rozvor nápravy, který je daný konstrukcí auta a limity natočení přední nápravy.

3.3.2 Ackermanova geometrie řízení

Za předpokladu, že obě přední kola přední nápravy jsou natočené o stejný úhel, jak bylo vypočteno v předcházející kapitole, došlo by ke smýkání kol. Důvodem smýkání jsou různé poloměry otáčení pro vnitřní a vnější kolo. Cílem je zajistit, aby se vnitřní i vnější kolo odvalovalo podle stejného poloměru, který platí pro model tříkolky. Toto lze zajistit při platnosti Ackermanovy podmínky.

Ackermanova podmínka:

Aby v zatáčce nedocházelo k prokluzu kol, musí být úhel natočení kol rozdílný – mají jeden společný střed zatáčení tzv. pól. [7]



Obr. 8: Ackermanova podmínka [7]

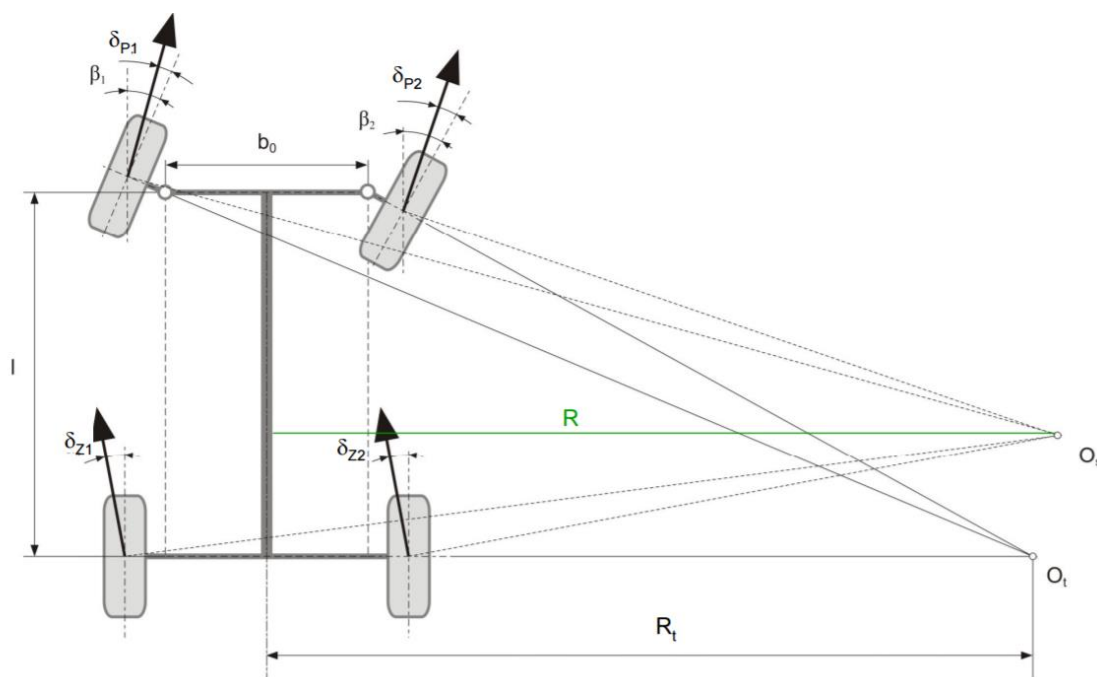
Matematicky lze vyjádřit Ackermannovou podmínku vztahem:

$$\cotg(|\beta_1|) - \frac{b_0}{2l} = \cotg(|\omega|) = \cotg(|\beta_2|) + \frac{b_0}{2l} \quad (6)$$

Kde, ω je natočení tříkolky ze vztahu (2). R_t je teoretický poloměr otáčení při zachování ideálních podmínek. β_1 je natočení vnitřního kola, β_2 natočení vnějšího kola.

Dynamika zatáčení

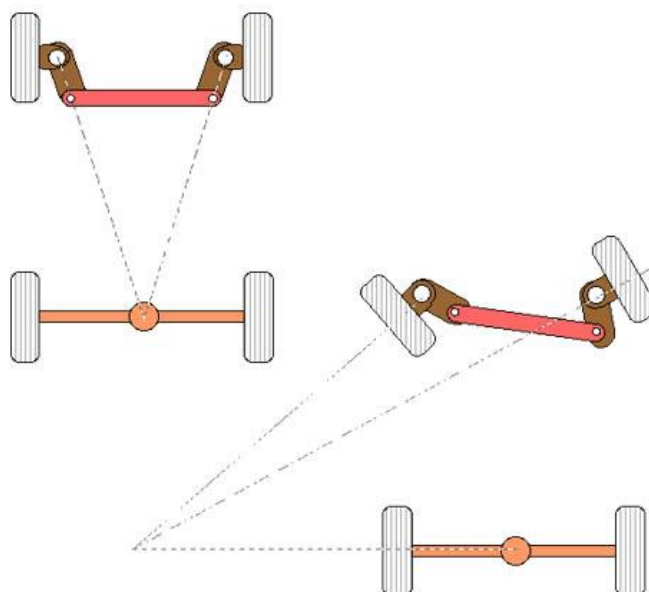
Při zatáčení vlivem boční síly a boční tuhosti pneumatik dochází k jevu, kdy zadní kola nejsou vždy dokonale rovnoběžné s osou vozidla.



Obr. 9: Skutečný model zatáčení [7]

Zde vidíme, že teoretický poloměr otáčení R_t není roven reálnému poloměru R . Pokud $R > R_t$, pak je auto nedotáčivé. V opačném případě je auto přetáčivé.

Způsob, jakým lze splnit Ackermannovou podmínku je klasický mechanismus řízení, konkrétně se jedná o kloubové lichoběžníkové řízení a převodovka řízení je realizována šnekovým či hřebenovým převodem.



Obr. 10: Zobrazení lichoběžníkového řízení

3.4 Porovnání lokalizačních metod

V kapitole 2.1.3 jsem uvedl lokalizační metodu, která byla použita v našem konvoji. Díky kameře získám pro svůj model několik bodů v čase, kudy projel vedoucí konvoj. Nevýhody odometrie jsou:

- nepřesnosti způsobené nerovným povrchem
- ztráta adheze
- nepřesnosti způsobené deformací kola
- větší než ideálně jednobodový kontakt kola s povrchem
- konečné rozlišení encodéru a konečná frekvence encodéru
- nedokonalosti v kruhovém tvaru kola

Tyto nevýhody naše metoda lokalizace nemá. Metoda lokalizace kamerou má ovšem své specifické nedostatky.

Hlavní je pozorovací úhel, který je na obě strany od osy kamery pouhých 15° . Také konečná frekvence snímku a nutnost dobrých světelných podmínek hraje určitou roli pro získání správných dat. Dále se o této problematice věnuje práce Patrika Kaury. [14]



Obr. 11: Značka

3.5 ROS (Robot Operating System)

ROS je v současnosti nejrozšířenější platformou pro robotické aplikace. Poskytuje mnoho knihoven a nástrojů, které usnadňují práci softwarovým developerům při tvorbě robotických aplikací. ROS je psán v C++ a je rozšířen po celém světě. Jeho služby využívají jak amatérské týmy, tak i profesionálové.

Nejnovější distribuce frameworku ROS z roku 2016 je ROS Kinetic Kane, která je primárně určena pro Ubuntu verze 16. ROS může využívat velké množství robotů. V našem případě lze pro lepší grafickou simulaci použít robota TurtleBot. Je zde také mnoho základních prvků pro řešení kinematických problémů, mnoho ovladačů pro různé typy senzorů a knihovny pro zpracování obrazu.

Výhoda použití frameworku ROS tkví v nezávislosti jednotlivých modulů na sobě a aplikace může fungovat i při nefunkčnosti jednoho modulu.

Příkladem nástroje vhodného pro simulaci je nástroj GAZEBO, který obsahuje programové rozhraní, ve kterém lze nasimulovat pohyb robota. Výhoda GAZEBA je, že se jedná o open-source. [8]

4 SIMULAČNÍ MODEL

V této části se práce zabývá simulačním modelem, který autor vytvořil v programovacím jazyce Python v programovém prostředí Pycharm. Cílem tohoto simulačního modelu bylo nalezení ideálního způsobu sledování vedoucího vozidla.

Principem konvoje je sledování vedoucího vozidla. Hlavním kritériem pro správné fungování konvoje je, aby si sledující auto nezkrátilo dráhu.

4.1 Princip fungování modelu

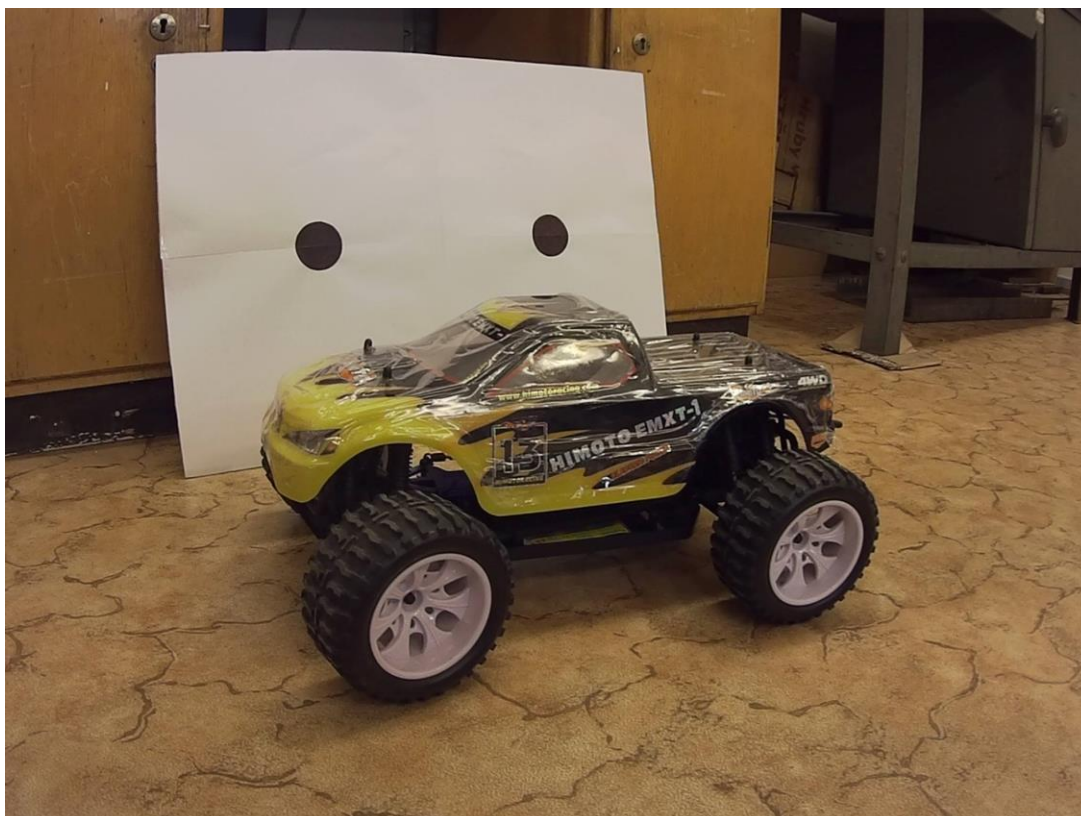
Úkolem sledujícího vozidla, dále označeno jako „slave vozidlo“, je co nejvěrněji projet trasu, kterou projelo vedoucí vozidlo, dále označeno jako „master vozidlo“. Algoritmus musí vyhodnocovat pozici mastera a poslat pokyn, jak má slave natočit kola, aby se dostal do zadané pozice. Vše by mělo probíhat real-time a pozice master vozidla je vypočtena k relativnímu postavení osy kamery slave vozidla.

Kvůli parametrům kamery, je nutné vzít v úvahu, že slave vozidlo se nesmí dostat moc blízko nebo moc daleko od master vozidla. Konkrétně se slave vozidlo od master vozidla musí pohybovat v rozsahu 200–600 mm. Což je další omezující podmínka pro sestavení sledovacího modelu.

Způsob, jak zajistit, aby si slave dráhu nezkracoval, je sestavení několika bodů, kudy master projel a poté projetí jednotlivých bodů. Tyto body jsou dále označovány jako „master body“.

4.2 Popis auta

Pro správný simulační model je naprosto klíčové co nejvěrněji napodobit reálnou situaci. Je tedy nutné znát konstrukční vlastnosti modelu. V bakalářském projektu byl použit model HIMOTO EMXT-1HI2101BL.



Obr. 12: Bugina HIMOTO

Na dalším obrázku jsou okótovány rozměry buginy. Pro simulační model řízení jsou důležité hlavně 2 parametry – rozvod kol a rozvor nápravy buginy.

Tyto parametry jsou důležité pro popis pohybu buginy, díky kterým pomocí vztahu (2) lze získat poloměr otáčení. Dále je velmi důležité znát maximální úhel kol. Čím větší je, tím lepší je manévrovatelnost. Model HIMOTO EMXT-1HI2101BL má maximální úhel natočení kol 15° .

Rychlost pohybu buginy také hraje velkou roli při sestavování simulačního modelu. Použitá bugina dokáže vyvinout maximální rychlost $0,5 \text{ m/s}$. Úhlová rychlost při natočení předního virtuálního kola o 15° je tedy $\omega = 0,52 \text{ rad/s}$, což odpovídá asi 30° za sekundu. Důvod, proč je zmíněno natočení virtuálního kola, je vysvětleno v kapitole 3.3.2. Stabilní jízdy lze dosáhnout při rychlostech $0,2 \text{ m/s}$.

$$dx = \operatorname{tg}(\phi) * s \quad (7)$$

$$dy = s \quad (8)$$

kde, dx a dy jsou změny polohy mastera v ose x a v ose y . Tyto vzorce ovšem platí za předpokladu, že osa kamery v počáteční pozici slave vozidla bude totožná s osou y .

V jiném než počátečním stavu je třeba pomocí rotační matice v eukleidovském prostoru najít pozici, protože globální kartézský systém se nepohybuje – pohybují se roboti v něm.

Rotace

Rotace v eukleidovské rovině je charakterizováno tím, že spojnice všech bodů s pevně zvoleným bodem, tzn. *středem otočení*, se změní o stejný úhel a vzdálenost bodů od středu otáčení zůstává nezměněná. Maticový zápis pro otáčení v rovině kolem počátku o definovaný úhel je zapsán v následující rovnici:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (9)$$

Kde x' , y' jsou souřadnice otočeného bodu, který měl před otočením souřadnice x , y . Úhel α je úhlem mezi osou kamery a osou y . [13]

Způsob, jakým by se dalo vylepšit sestavení master bodů je přes aproximaci, díky které lze získat více master bodů. Čím více bodů, tím lepší jsou informace o pohybu mastera. Při znalosti více bodů, kudy master vozidlo z největší pravděpodobnosti projelo, lze dosáhnout správného nadjíždění a téměř dokonalého sledování trasy master vozidla. Také díky aproximaci lze kompenzovat krátkodobé výpadky kamery. Z možných metod aproximací je nejvýhodnější metoda nejmenších čtverců, která počítá s nepřesnostmi měření, a jelikož má kamera toleranci měření okolo 5 mm, tak je tato metoda vhodná pro simulační model, nevýhodou je však větší softwarová náročnost výpočtu.

4.4 Řízení slave vozidla

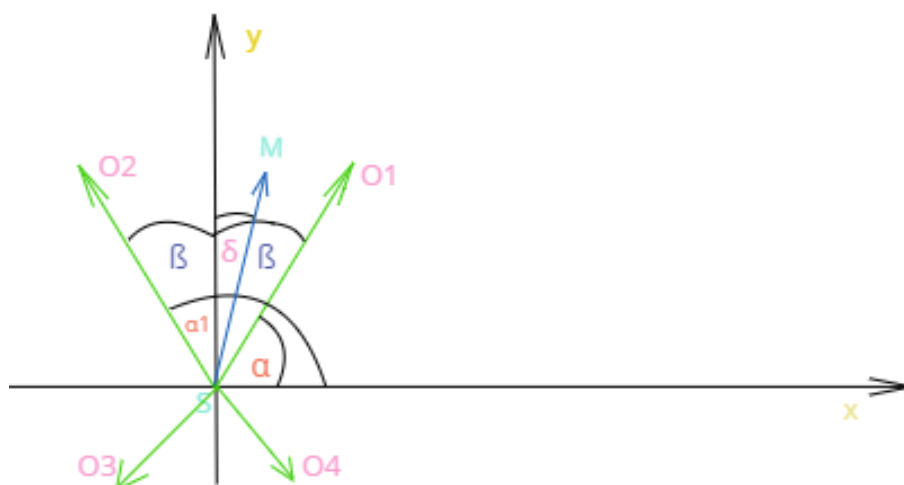
Pohyb vozidla je křivkou složenou z mnoha navazujících kružnic. Jízda rovně je pohyb po kružnici s nekonečným poloměrem. V každém uživateli definovaném časovém intervalu se robot rozhodne, jakým způsobem natočí kola a po vzniklé kružnici ujede určitou vzdálenost. Tento časový interval je lépe popsán ve fázi 2. Algoritmus postupuje v těchto 5 fázích:

1. rozhodnutí, zda jet doprava nebo doleva
2. výpočet úhlu mezi master a slave vozidlem

3. výpočet natočení kola pomocí PI regulace
4. transformace souřadnic
5. opakuje se bod 1, dokud nejsem u cíle

4.4.1 Fáze 1

Rozhodnutí, zda pojedou doleva nebo doprava závisí na postavení v kartézském souřadném systému a ose kamery. Úhel mezi dvěma vektory jsem přepočítal přes skalární součin, jehož výstupem je absolutní hodnota úhlu. Řešení je ukázáno na následujícím schématu.



Obr. 14: Schéma

S , M , jsou možné pozice vozidel. Z analytické geometrie lze získat vektor $|\overrightarrow{SM}|$, který je na obrázku vyobrazen jako modrá šipka. V kapitole 4.3 je vysvětleno, jak byl zjištěn bod M . Ve zkratce, je to bod kudy projelo master vozidlo a v ideálním případě jim projede i slave vozidlo.

V této fázi je cílem zjistit, jakým způsobem je nakloněna osa kamery, na obrázku zobrazena jako zelená šipka. Je potřeba zjistit, v jakém leží kvadrantu a jaký je úhel mezi ní a vektorem $|\overrightarrow{SM}|$. Pomocí skalárního součinu zjistím úhly α a β a tím pádem i postavení v souřadném systému. Jelikož zjišťuji absolutní hodnotu úhlu mezi dvěma vektory, potřebuji pro správnou lokalizaci 2 úhly.

- první kvadrant: $0 \leq \alpha \leq 90^\circ \wedge 0 \leq \beta \leq 90^\circ$
- druhý kvadrant: $90^\circ \leq \alpha \leq 180^\circ \wedge 0 \leq \beta \leq 90^\circ$
- třetí kvadrant: $180^\circ \leq \alpha \leq 270^\circ$
- čtvrtý kvadrant: $270^\circ \leq \alpha \leq 360^\circ$

Následně je třeba zjistit, jestli zatočit doleva nebo doprava. K tomu je nutné porovnat velikosti úhlu δ s úhlem β . Pokud se nachází osa kamery ve třetím, popřípadě

čtvrtém kvadrantu, je třeba úhel δ modifikovat. V tomto případě se bude jednat o úhel mezi master vozidlem a *záporným* směrem osy y .

4.4.2 Fáze 2

Tato fáze vypočítává momentální úhel mezi master a slave vozidlem. V reálné situaci toto není potřeba, jelikož tuhle činnost zajišťuje kamera. V simulaci ovšem logicky kamera není k dispozici, tudíž přepočít momentálního úhlu je proveden pomocí standardních goniometrických funkcí.

Pro výpočet se vypočte master a slave vektor. Algoritmus si tedy pamatuje dvě poslední slave vozidla. Master vektor se zjistí z aktuální pozice slave vozidla a bodu, ke kterému slave vozidlo směřuje. V počátečním stavu je u slave vozidla vektor shodný s kladným směrem osy y . *Slave vektor* – Koncový bod aktuální pozice slave vozidla. Počáteční bod předchozí pozice slave vozidla.

4.4.3 Fáze 3

V této kapitole popsán způsob výpočtu trajektorie slave vozidla a následná míra natočení pomocí PI regulátoru.

Nejprve se definuje pohyb. V tomto případě se bude jednat o rovnoměrný pohyb po kružnici. Toto omezení není tak závažné, jelikož aktualizace výpočtu simulace se děje každé 0,2 sekundy. Tento časový interval byl zvolen z empirických dat, které autor nasbíral testováním různých trajektorií pohybu vozidel. Rychlost je taktéž konstantní 0,2 m/s. Zde je prostor pro zlepšení algoritmu, jelikož různá rychlost může lépe přizpůsobit trajektorii slave vozidla. V našem případě jsou vzdálenosti a velikosti rychlosti malé, proto regulace rychlosti byla nadbytečná.

Základní rovnice, která určuje pohyb bodu po kružnici je:

$$\varphi = \frac{v}{R} * t \quad (10)$$

kde φ je úhlová dráha. Rychlost v a čas t jsme si popsali výše. Poloměr R lze zjistit pomocí rovnice (2). V rovnici (2) je třeba dosadit ω , která je úhlem natočení kol přední nápravy.

Pro natočení předních kol nápravy je třeba si uvědomit její limity. V určitých situacích je úhel příliš vysoký, tudíž je třeba ho omezit na 15°. Dále je tento úhel, který vychází z rozdílu úhlu mezi master a slave vektorem upraven PI regulátorem, který je nastaven pomocí Ziegler-Nicholsové metody.

4.4.4 Fáze 4

V této části se úhlová dráha transformuje do souřadného systému. K tomu budou použity rovnice (4) a (5), kde za úhel φ se dosadí úhlová dráha, kterou slave vozidlo

projelo za uživatelem definovaný časový úsek. Je třeba následně změnu polohy otočit o úhel α , který se rovná všem úhlovým dráhám pohybu slave vozidla mínus úhlová dráha posledního úseku. K této činnosti použijí rovnici (9). Pro lepší pochopení ji uvádím v rozepsaném tvaru:

$$x' = x * \cos(\alpha) - y * \sin(\alpha) \quad (11)$$

$$y' = x * \sin(\alpha) + y * \cos(\alpha) \quad (12)$$

kde α je úhel rotace a x', y' jsou souřadnice po rotaci.

4.4.5 Fáze 5

Pro lepší zaoblení trajektorie slave vozidla jsem se rozhodl, že jakmile se slave vozidlo dostane do dostatečné blízkosti bodu, kterým má projet, přepne svou pozornost na další bod v pořadí. Této vlastnosti dosáhnu pomocí vzorce pro kružnici:

$$x^2 + y^2 \leq r^2 \quad (13)$$

Jakmile bude platit rovnice (13), slave vozidlo bude následovat v pořadí následující master bod. V mém simulačním modelu je $r = 0,1$ m, který byl odvozen z empirického pozorování.

4.5 Testování

Testování je asi nejdůležitější pasáží praktické části této bakalářské práce. V této kapitole budou popsány jednotlivé simulační trasy s odkazy na jednotlivé fáze.

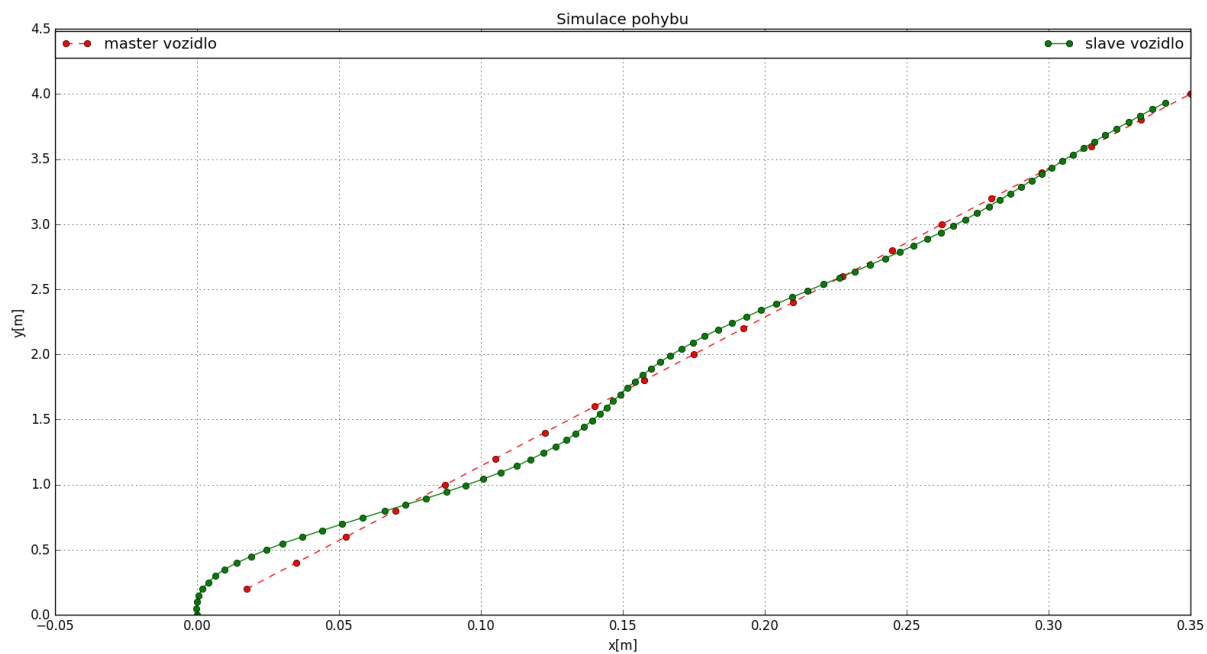
Rovný pohyb

Nejprve se vyzkouší systém při sledování rovného pohybu. Aktualizace polohy se díky kameře pohybuje 25krát za sekundu, rozdíl mezi body tedy bude velmi malý. Na této simulaci si demonstrováme proporcionalní složku našeho regulátoru.

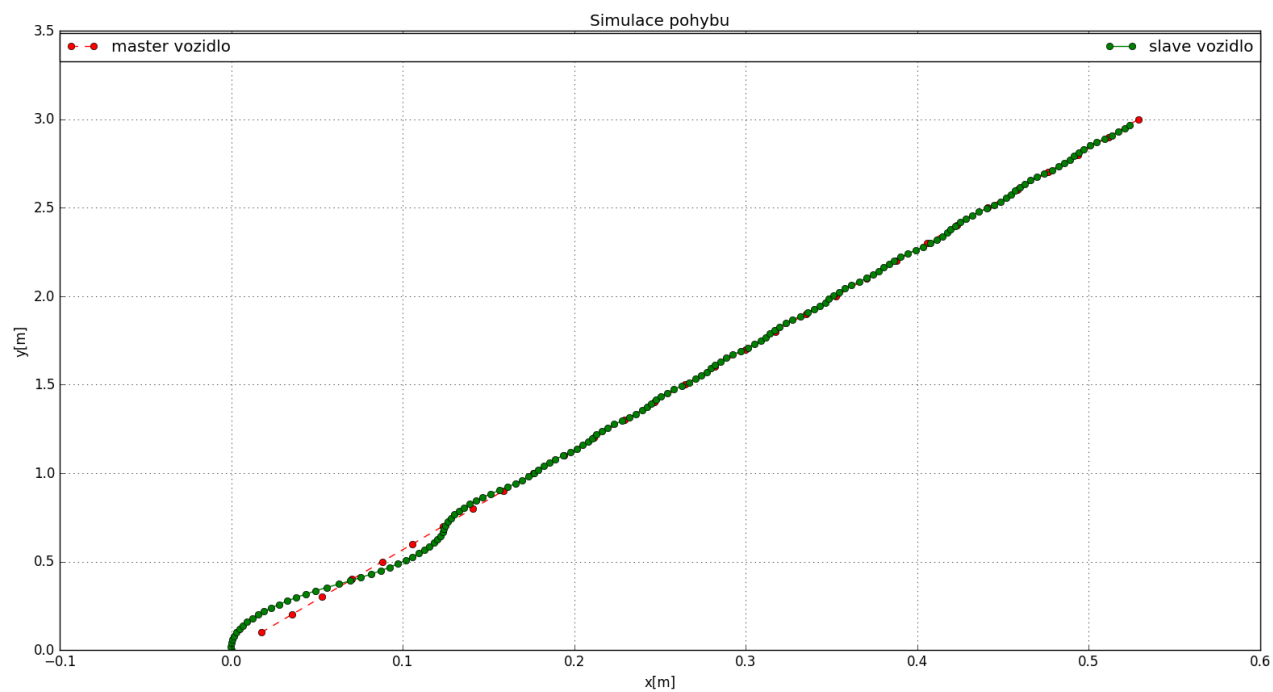
Na obrázku 15. jsem nastavil regulátor schválně na nízkou hodnotu. Slave vozidlu trvá déle, než si zkoriguje směr se směrem master vozidla. Maximální odchylka je cca 60 cm, což je nežádoucí.

Na obrázku 16. je nastavena proporcionalní složka regulátoru schválně na vysokou hodnotu. Je zde vidět zlepšení oproti obrázku číslo 15, ovšem detailní pohled nám odhalí velmi nepříjemné malé odchylky, kdy regulátor nutí neustále lehce korigovat zatáčení, což není vhodné z důvodu opotřebení součástí (obrázek 17).

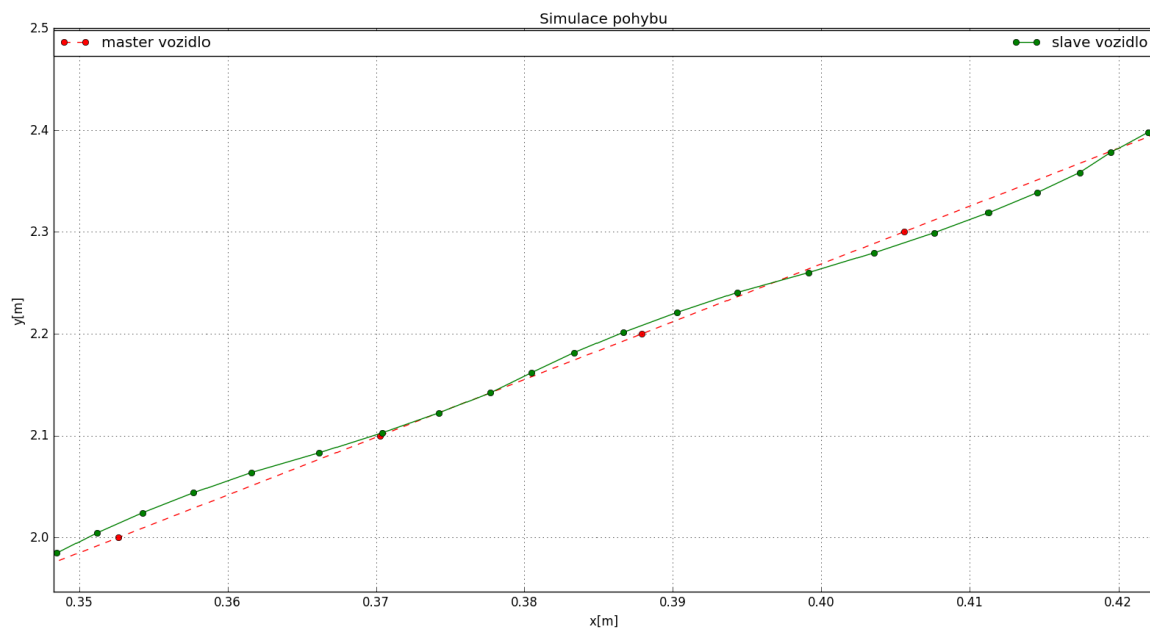
Obrázek 18. a 19. ukazuje ideální nastavení PI regulátoru. Jak si můžeme všimnout, není zde zbytečné korekce malých odchylek, nedochází tedy k oscilaci křivky pohybu. Maximální odchylka dráhy slave vozidla od ideální dráhy master vozidla je cca 5 cm, což je v rámci tolerance chyby měření.



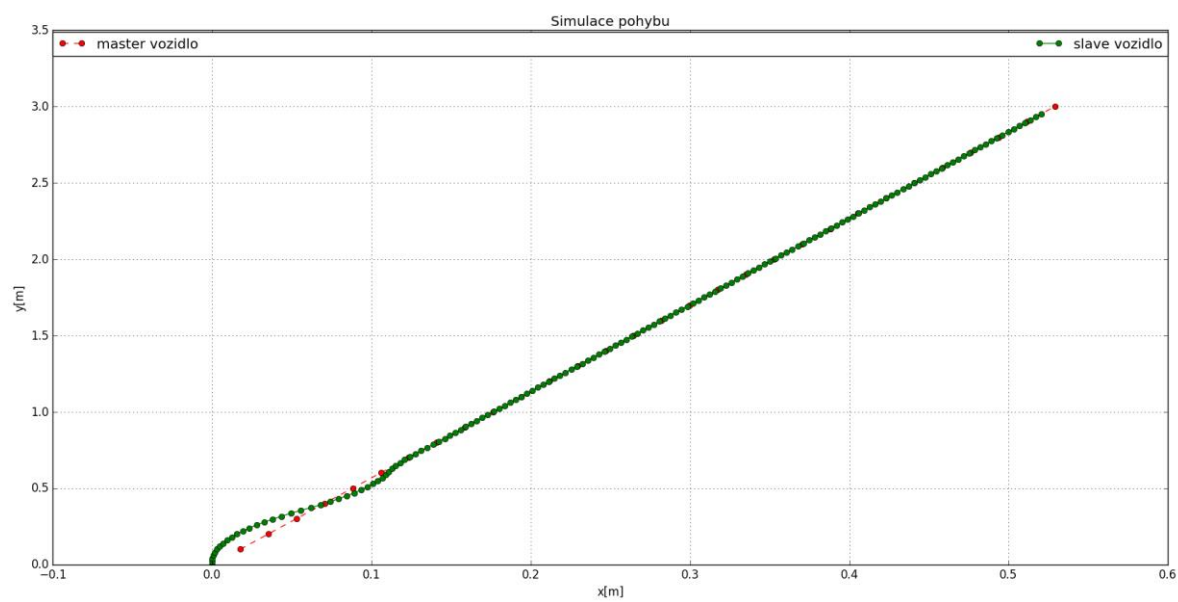
Obr. 15: Jízda rovně nízká hodnota P složky



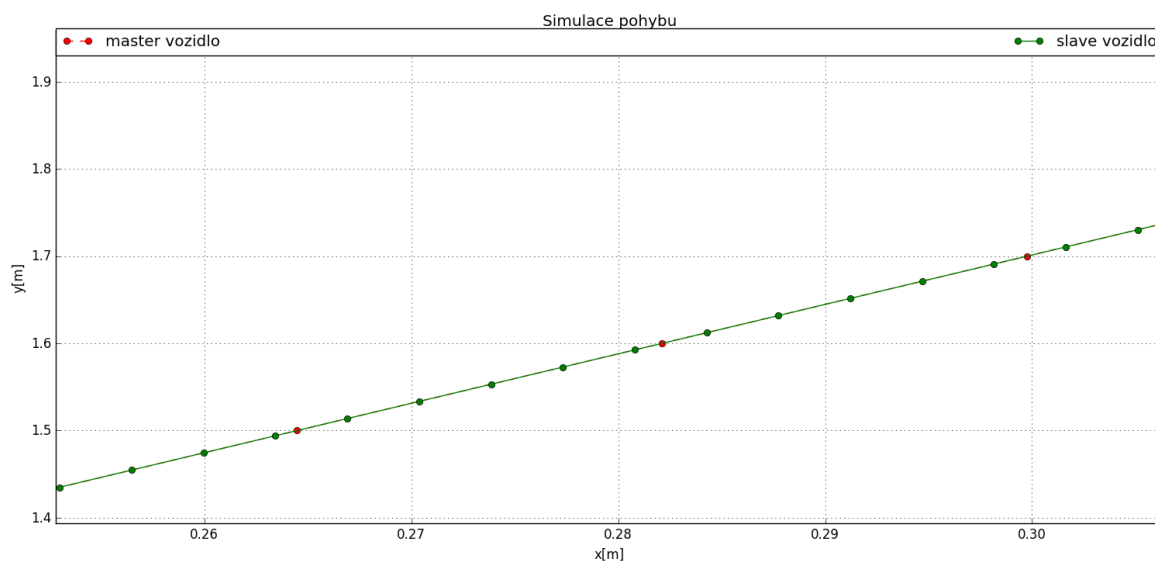
Obr. 16: Jízda rovně vysoká hodnota P složky



Obr. 17: Detail pohybu jízdy s vysokou hodnotou P složky



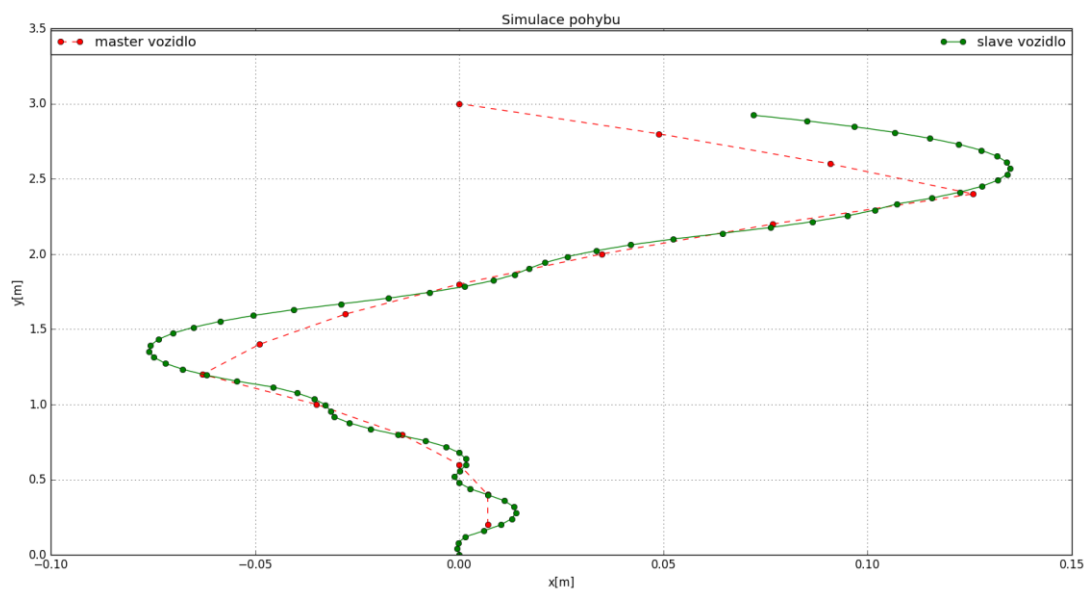
Obr. 18: Jízda s dobře nastaveným PI regulátorem



Obr. 19: Detail jízdy s ideálně nastaveným regulátorem

Uživatelský pohyb

Z rovnic (2) a (10) lze zjistit, že master vozidlo může ujet za 0,2 sekundy maximální úhlovou dráhu $2,23^\circ$. Samozřejmě, při stejných konstrukčních specifikacích jako slave vozidlo. Pokud se tedy model chová realisticky, nesmí master body být příliš od sebe. Při příliš blízké vzdálenosti slave vozidla od master bodu se slave vozidlo bude soustředit na následující bod. Maximální odchylka od ideální trajektorie je cca 30cm.



Obr. 20: Uživatelský pohyb

5 ZÁVĚR

Zadáním bakalářské práce bylo vypracování rešeršní studie zabývající se současnými metodami autonomního řízení vozidla. V praktické části byl sestaven a otestován simulační model.

V teoretické části byly popsány nejnovější trendy z hlediska autonomního řízení vozidel a jako příklad byl uveden autonomní systém Enhanced Autopilot od firmy Tesla. Našemu konvoji odpovídá projekt SARTE, který má své využití v nákladní dopravě.

V praktické části za pomoci vzorců o rovnoměrném pohybu po kružnici, analytické geometrii a transformaci souřadnic byl vytvořen simulační model, jehož cílem bylo zjištění proporcionální a integrační složky pro regulaci natočení kol vozidla. Dále byl navržen algoritmus, který si ukládá pozice vedoucího vozidla a těmito pozicemi projíždí i následující vozidla v konvoji. Díky tomuto algoritmu nedochází ke zkracování dráhy vozidel v konvoji.

6 SEZNAM POUŽITÉ LITERATURY

- [1] Statistika nehodovosti. *Policie ČR* [online]. Praha: Policie, 2017 [cit. 2017-05-03]. Dostupné z: <http://www.policie.cz/clanek/statistika-nehodovosti>
- [2] Tesla Autopilot. *Tesla* [online]. Palo Alto, Kalifornie: Musk, c2011-2016 [cit. 2017-05-02]. Dostupné z: <https://www.tesla.com/autopilot>
- [3] How does LiDAR work. *Lidar-uk* [online]. Coalville: Bluesky International Limited, 2017 [cit. 2017-05-05]. Dostupné z: <http://www.lidar-uk.com/how-lidar-works/>
- [4] What is Ultrasonic Sensor. *What is Ultrasonic Sensor* [online]. [cit. 2017-05-05]. Dostupné z: http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html
- [5] Prostředky relativní lokalizace. *Srovnání lokalizačních technik* [online]. [cit. 2017-05-05]. Dostupné z: <http://marek.sk.sweb.cz/lokalizace.html>
- [6] WINKLER, Zbyněk. *Robotika.cz* [online]. 2005-05-12 [cit. 2017-06-02]. Odometrie. Dostupné z <https://robotika.cz/guide/odometry/cs>.
- [7] Systémy směrového řízení. *Česká asociace pro hydrauliku a pneumatiku* [online]. Liberec, 2013 [cit. 2017-05-07]. Dostupné z: <http://www.cahp.cz/wp-content/2013/03/02-Systemy-smeroveho-řízení.pdf>
- [8] ROS.org ROS.org[Powering the world robots. [online]. Stanford: Willow Garage, 2017 [cit. 2017-05-08]. Dostupné z: <http://www.ros.org/>
- [9] Princip letu. *Droneweb* [online]. Praha, c2016 [cit. 2017-05-09]. Dostupné z: <http://www.droneweb.cz/konstrukce/item/42-kvadroptera-řízení>
- [10] HIMOTO. *himotoracing.com* [online]. Taiwan China, 2016 [cit. 2017-05-09]. Dostupné z: http://www.himotoracing.com/product/truck/electric/truck_e004/details.php?information=0
- [11] ROBERTS, Wesley. *DARPA releases video of new-and-improved LS3 quadruped robots* [online]., 1 [cit. 2017-05-11]. Dostupné z: <http://newatlas.com/new-ls3-quadruped-robot/24098/>
- [12] HUENERMANN, Jan. Self-driving cars in the browser. Self-driving cars in the browser [online]. 2016 [cit. 2017-05-11]. Dostupné z: <http://janhuenermann.com/projects/learning-to-drive>
- [13] Geometrické transformace. *Matematika online VUT* [online]. Brno: Akademické nakladatelství CERM, 2006 [cit. 2017-05-13]. Dostupné z: http://mathonline.fme.vutbr.cz/pg/Algoritmy/08_TRANSFORMACE.htm
- [14] KAURA, P. Návrh a realizace detekce a sledování značky pro účel řízení semiautonomního konvoje. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 60 s. Vedoucí bakalářské práce Ing. Michal Růžička.
- [15] BUCHTA, H. Návrh struktury řídicího systému pro konvoj autonomních vozidel s využitím frameworku ROS. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 41 s. Vedoucí bakalářské práce doc. Ing. Stanislav Věchet, Ph.D.
- [16] How does a military tank turn. <https://www.quora.com/How-does-a-military-tank-turn-left-or-right> [online]. 2016 [cit. 2017-05-21]. Dostupné z: <https://www.quora.com/How-does-a-military-tank-turn-left-or-right>

7 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

Slave vozidlo – autonomní auto sledující vedoucí vozidlo

Master vozidlo – vedoucí vozidlo

Master bod – bod, který vznikl sledováním trajektorie master vozidla

Master vektor – Koncový bod aktuální master bod. Počáteční bod aktuální pozice slave vozidla

Slave vektor – Koncový bod aktuální pozice slave vozidla. Počáteční bod předchozí pozice slave vozidla

Seznám obrázků

Obr. 1: Vývoj počtu usmrcených osob za leden od roku 1980	17
Obr. 2: Popis senzorů v autopilotu Tesly	18
Obr. 3: Lidar SICK LMS100 – 100000	19
Obr. 4: Schéma konvoje	20
Obr. 5: Robot LS3 od firmy DARPA	23
Obr. 6: Princip pohybu dronu	25
Obr. 7: Model tříkolky	26
Obr. 8: Ackermanova podmínka	27
Obr. 9: Skutečný model zatáčení	28
Obr. 10: Zobrazení lichoběžníkového řízení	29
Obr. 11: Značka	30
Obr. 12: Bugina HIMOTO	32
Obr. 13: Rozměry buginy v mm	33
Obr. 14: Schéma	35
Obr. 15: Jízda rovně nízko hodnota P složky	38
Obr. 16: Jízda rovně vysoká hodnota P složky	38
Obr. 17: Detail pohybu jízdy s vysokou hodnotou P složky	39
Obr. 18: Jízda s dobře nastaveným PI regulátorem	39
Obr. 19: Detail jízdy s ideálně nastaveným regulátorem	40
Obr. 20: Uživatelský pohyb	40